**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 6**

*дисциплина:* ***Архитектура вычислительных систем***

Студент: Панченко Денис Дмитриевич

Группа: НБИбд-03-22

**МОСКВА**

2022 г.

**Цель работы:**

Освоить арифметические инструкции языка ассемблера NASM.

**Ход работы:**

1) Создаем каталог для программ лабораторной работы № 6, переходим в

него и создаем файл lab6-1.asm. (Рис. 1)



Рис. 1. Создание каталога с файлом

2) Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. (Рис. 2)  
Создаем исполняемый файл и запускаем его. (Рис. 3)

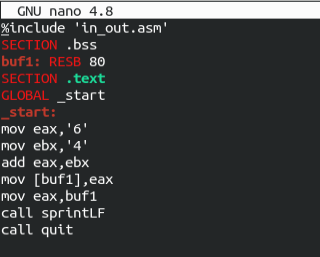


Рис. 2. Текст программы

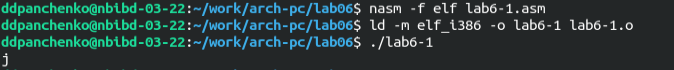


Рис. 3. Запуск файла

3) Далее изменяем текст программы и вместо символов, записываем в регистры числа. Исправляем текст программы. (Рис. 4)

Создаем исполняемый файл и запускаем его. (Рис. 5)

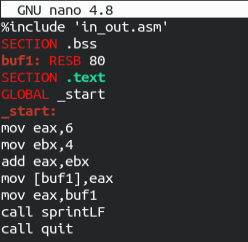


Рис. 4. Текст программы

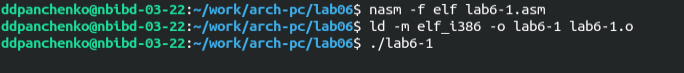


Рис. 5. Запуск файла

Этот символ не отображается при выводе на экран.

4) Создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него

текст программы. (Рис. 6-7)  
Создаем исполняемый файл и запускаем его. (Рис. 8)



Рис. 6. Создание файла

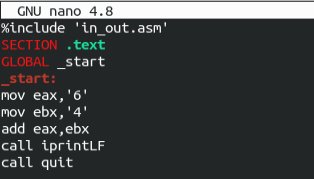


Рис. 7. Текст программы

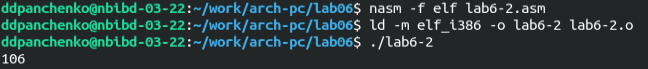


Рис. 8. Запуск файла

5) Аналогично предыдущему примеру изменяем символы на числа. (Рис. 9)

Создаем исполняемый файл и запускаем его. (Рис. 10)

Заменяем функцию iprintLF на iprint. Создаем исполняемый файл и запускаем его. (Рис. 11-12)

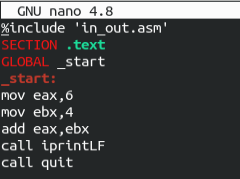


Рис. 9. Текст программы

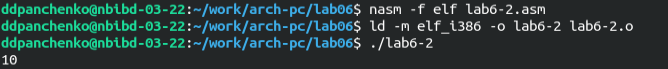


Рис. 10. Запуск файла

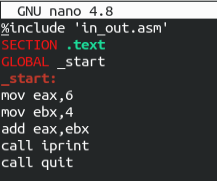


Рис. 11. Текст программы

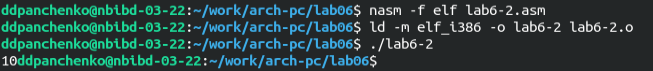


Рис. 12. Запуск файла

Разница в том, что подпрограмма iprintLF переводит сообщение на следующую

строку, а iprint данного действия не делает.

6) Создаем файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 и вводим в него

текст программы. (Рис. 13-14)

Создаем исполняемый файл и запускаем его. (Рис. 15)

Изменяем текст программы для вычисления выражения 𝑓(𝑥)=(4∗6+2)/5. (Рис. 16)

Создаем исполняемый файл и проверяем его работу. (Рис. 17)



Рис. 13. Создание файла

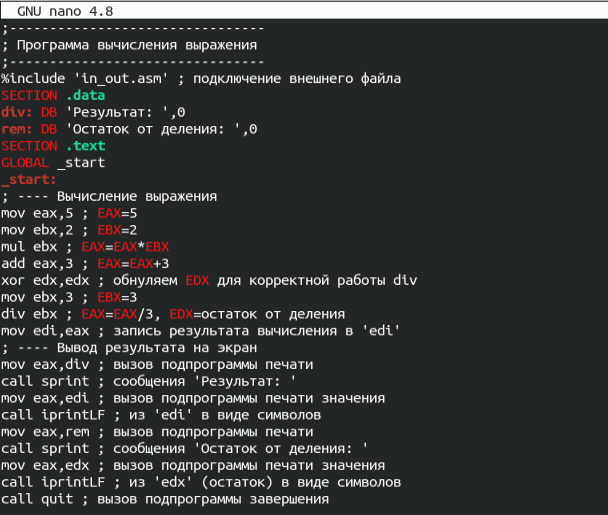


Рис. 14. Текст программы

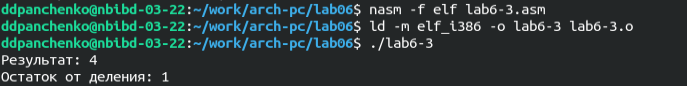


Рис. 15. Запуск файла

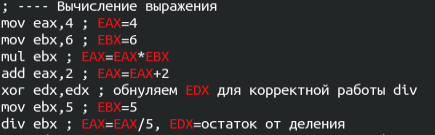


Рис. 16. Изменения

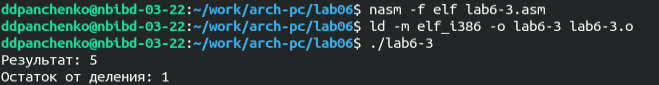


Рис. 17. Запуск файла

7) Создаем файл variant.asm в каталоге ~/work/arch-pc/lab06 и вводим в него

текст программы. (Рис. 18-19)

Создаем исполняемый файл и запускаем его. (Рис. 20)



Рис. 18. Создание файла

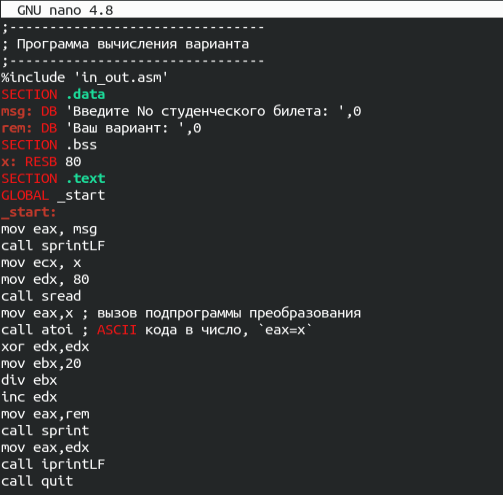


Рис. 19. Текст программы

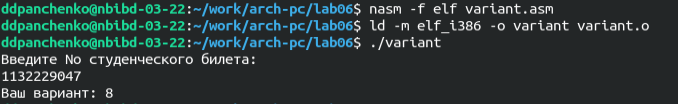


Рис. 20. Запуск файла

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

mov eax,rem   
call sprint

2. Для чего используется следующие инструкции nasm?   
mov ecx,x – запись входной переменной в регистр ecx;

mov edx,80 – запись переменной в регистр edx;

call spread – вызов процедуры чтения данных.

3. Для чего используется инструкция “call atoi”?

Для вызова подпрограммы преобразования ASCII кода в число, `eax=x`.

4. Какие строки листинга 7.4 отвечают за вычисления варианта?

xor edx,edx

mov ebx,20

div ebx

inc edx.

5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

В регистр ebx.

6. Для чего используется инструкция “inc edx”?

Для увелечения на 1.

7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений?

mov eax,rem

call sprint

mov eax,edx

call iprintLF.

**Задание для самостоятельной работы.**

1) Напишем программу вычисления выражения 𝑦 = (11 + 𝑥) ⋅ 2 – 6 (Вариант 8). (Рис. 21)

Получаем исполняемый файл и проверяем его работу для значений x1 = 1 и x2 = 9 (Вариант 8). (Рис. 22)

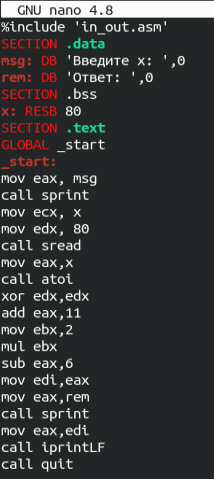


Рис. 21. Текст программы

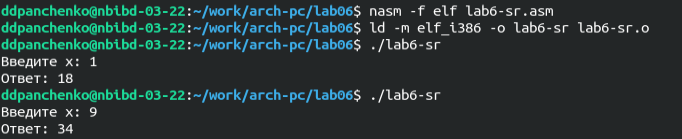


Рис. 22. Запуск файла

**Вывод:**

В ходе выполнения данной лабороторной работы я освоил арифметические инструкции языка ассемблера NASM.